

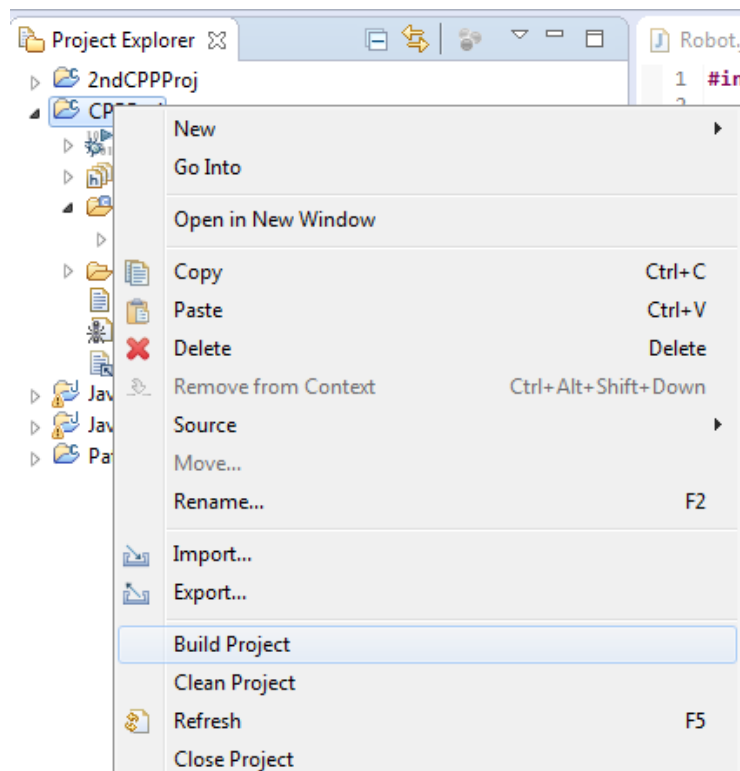
# Building and downloading a robot project to the roboRIO

There are two ways of running programs onto the roboRIO. You can

1. Attach to the roboRIO and run the program using the debugger from your development system OR
2. Load it onto the roboRIO flash drive so it will run on reboot.

For tournaments you should always download the program so that it will be there when the robot is restarted and the match is played. This article will cover loading the program onto the roboRIO to run on reboot. The next article will cover the debugger.

## Building a robot program

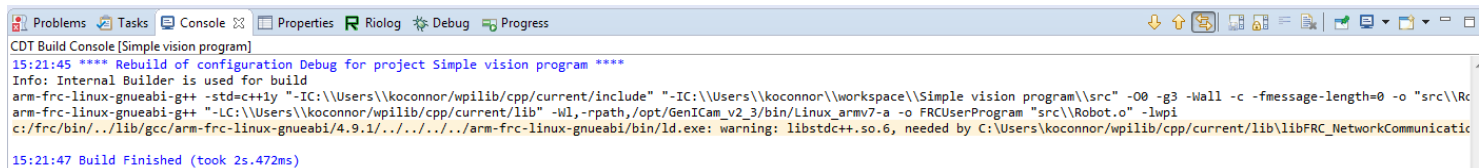


The first step to getting the program on the roboRIO is to build the code. This will compile and link the project files. If you have set the option to build automatically as recommended in the Eclipse installation article, this step is done each time you save a file in the project. If not, or if you just

# Building and downloading a robot project to the roboRIO

want to be sure, you can build the project by right clicking on it in the Project Explorer (left pane) and selecting Build Project

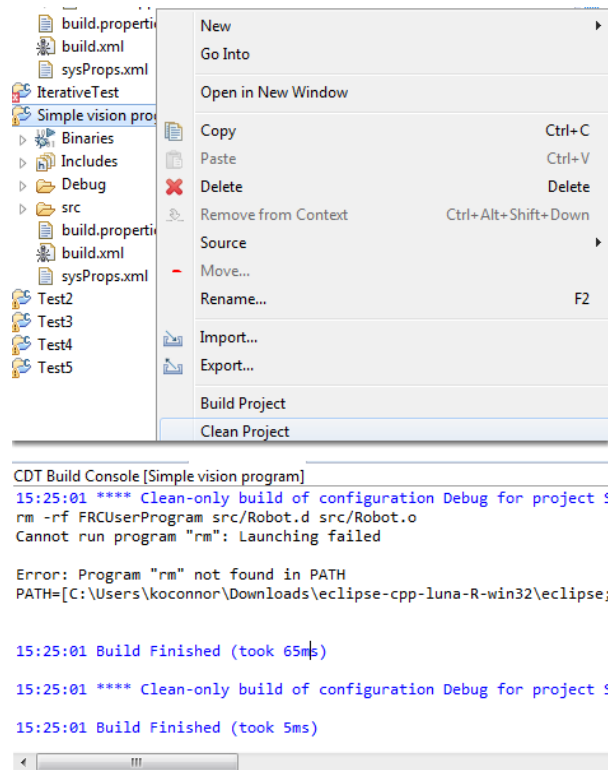
## Build Console



```
CDT Build Console [Simple vision program]
15:21:45 ***** Rebuild of configuration Debug for project Simple vision program *****
Info: Internal Builder is used for build
arm-frc-linux-gnueabi-g++ -std=c++1y "-IC:\Users\koconnor\wpilib\cpp\current\include" "-IC:\Users\koconnor\workspace\Simple vision program\src" -O0 -g3 -Wall -c -fmessage-length=0 -o "src\Robot.o" "src\Robot.cpp"
arm-frc-linux-gnueabi-g++ "-LC:\Users\koconnor\wpilib\cpp\current\lib" -Wl,-rpath,/opt/GenICam_v2_3/bin/Linux_armv7-a -o FRCUserProgram "src\Robot.o" -lwpilib
c:/frc/bin/./lib/gcc/arm-frc-linux-gnueabi/4.9.1/../../../../arm-frc-linux-gnueabi/bin/ld.exe: warning: libstdc++.so.6, needed by C:\Users\koconnor\wpilib\cpp\current\lib\libFRC_NetworkCommunicatio
15:21:47 Build Finished (took 2s.472ms)
```

A successful build will look similar to the image above. Note the warning in yellow, this warning is harmless and will occur each time you try to build your project, it can be safely ignored.

## Clean

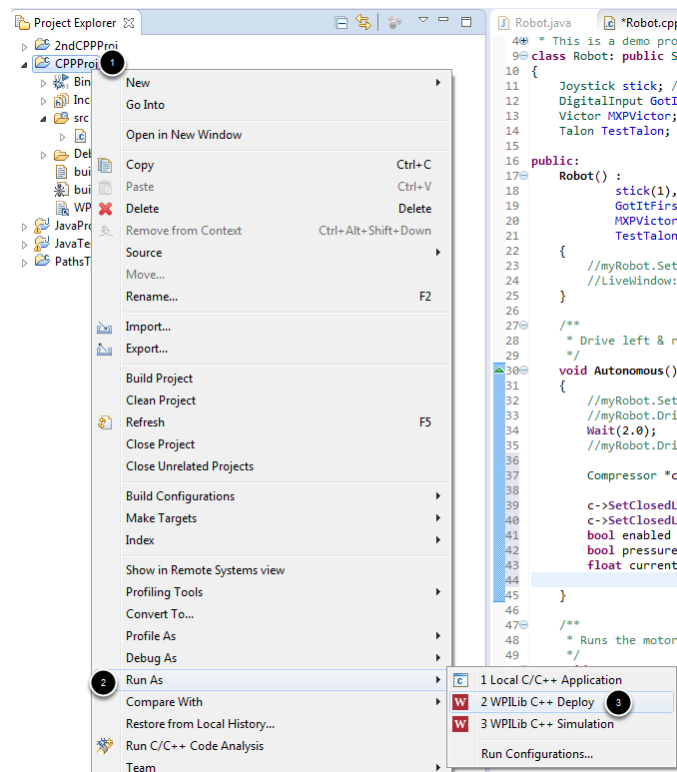


If you want to be absolutely sure your project is re-built, you may wish to do a Clean before building. Right click on the project and select "Clean Project". This will delete the build products for your project.

# Building and downloading a robot project to the roboRIO

Note the errors shown in the image above. These errors are normal and do not indicate that the Clean did not complete successfully. To verify that the Clean succeeded, make sure that the "Debug" directory no longer shows up under the Project in the Project Explorer (the left pane of the Eclipse window).

## Loading a program to run on robot startup



To have the program run automatically each time the roboRIO boots, it must be loaded in the roboRIO flash memory. For tournaments you should always download the program so that it will be there when the robot is restarted and the match is played. To do this:

1. Right click on the project in the Project Explorer
2. Select "Run As"
3. Select "WPILib C++ Deploy"

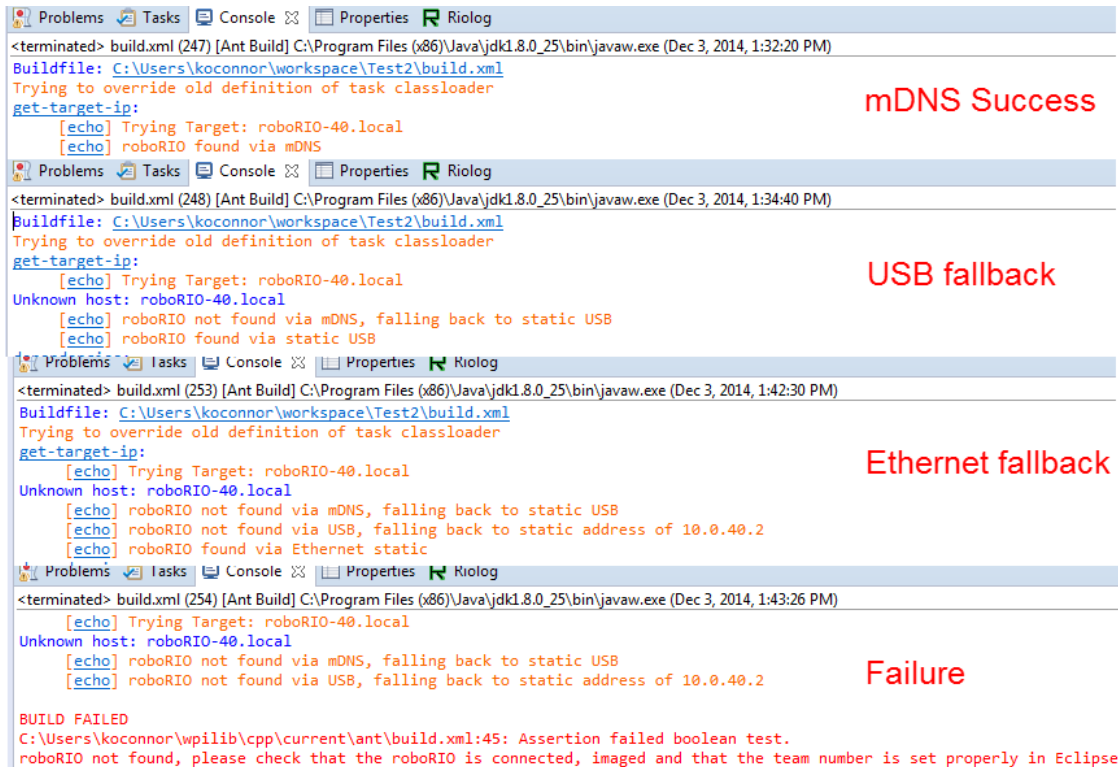
The deployment process will then begin and start outputting status information to the Console (generally located in the bottom center or bottom right of the window). After the program is downloaded it will be immediately launched, there is no need to reboot the roboRIO.

# Building and downloading a robot project to the roboRIO

## Program Download Process

After you click the WPILib C++ Deploy button the deploy process will proceed through a number of steps.

### Get Target IP



```
<terminated> build.xml (247) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:32:20 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
[echo] roboRIO found via mDNS
mDNS Success

<terminated> build.xml (248) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:34:40 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO found via static USB
USB fallback

<terminated> build.xml (253) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:42:30 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO not found via USB, falling back to static address of 10.0.40.2
[echo] roboRIO found via Ethernet static
Ethernet fallback

<terminated> build.xml (254) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:43:26 PM)
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO not found via USB, falling back to static address of 10.0.40.2
Failure

BUILD FAILED
C:\Users\koconnor\wpilib\cpp\current\ant\build.xml:45: Assertion failed boolean test.
roboRIO not found, please check that the roboRIO is connected, imaged and that the team number is set properly in Eclipse
```

In this step, the plugin determines the IP of the roboRIO. The plugin will try to reach the roboRIO via the following methods, in order:

1. mDNS (roboRIO-TEAM.local where TEAM is your team number)
2. USB static IP (172.22.11.2)
3. Ethernet static IP (10.TE.AM.2)

If the roboRIO is accessible via any of these methods, the deploy will proceed using that address as the target. If the roboRIO cannot be found using any of these methods, the deploy will abort. See [Connection Failure in the Troubleshooting](#) section below.

# Building and downloading a robot project to the roboRIO

## Dependencies

```
buildfile: C:\Users\koconnor\workspace\1est2\build.xml
Trying to override old definition of task classloader
get-target-ip:
    [echo] Trying Target: roboRIO-40.local
    [echo] roboRIO found via mDNS
dependencies:
    [echo] roboRIO image version validated
deploy:
get-target-ip:
    [echo] Trying Target: roboRIO-40.local
    [echo] roboRIO found via mDNS
dependencies:
```

**Image OK**

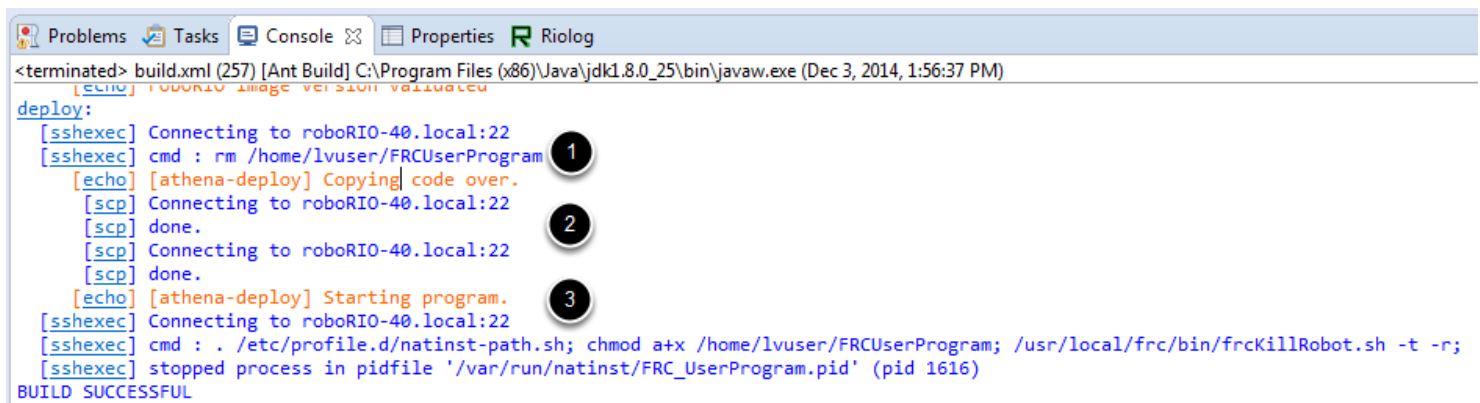
---

**Image mismatch**

**BUILD FAILED**  
C:\Users\koconnor\wpilib\cpp\current\ant\build.xml:119: Assertion failed boolean test.  
roboRIO Image does not match plugin, allowed image version: 19

The dependencies section checks that the roboRIO contains any dependencies necessary for operation. For C++ at this time, this consists of a check of the roboRIO image version. If the validation succeeds you will get a message that the version was validated and the deploy will proceed. If the check fails, you will see a message indicating the allowed image version for this version of the plugins, see [Image Check Failure under Troubleshooting](#) below for more details

## Copy and Start code



```
<terminated> build.xml (257) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:56:37 PM)
[echo] roboRIO image version validated
deploy:
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : rm /home/lvuser/FRCUserProgram
[echo] [athena-deploy] Copying code over.
[scp] Connecting to roboRIO-40.local:22
[scp] done.
[scp] Connecting to roboRIO-40.local:22
[scp] done.
[echo] [athena-deploy] Starting program.
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : . /etc/profile.d/natinst-path.sh; chmod a+x /home/lvuser/FRCUserProgram; /usr/local/frc/bin/frcKillRobot.sh -t -r;
[sshexec] stopped process in pidfile '/var/run/natinst/FRC_UserProgram.pid' (pid 1616)
BUILD SUCCESSFUL
```

The remaining parts of the deploy then proceed as follows:

1. Delete existing program. This is required because you cannot perform an overwrite while the program is running. If there is no existing program you will see a red error message under this line, but the deploy will proceed.
2. Copy user program and robotCommand file. The robotCommand file tells the framework that launches the code what to launch (C++ vs. Java and Debug vs deploy).

# Building and downloading a robot project to the roboRIO

3. Start code. The plugin calls a script file which kills the existing user program, then restarts the LabVIEW RT process. This starts the code in the same way it will be launched when the roboRIO is booted.

## Deployed Code

There is no need to reboot the roboRIO to run the new code, it will begin running automatically. After the deploy completes, each time the roboRIO is rebooted it will run the deployed code until new code (from any FRC language) is downloaded.

## Viewing program output

The console output from the program does not appear in the regular Eclipse console. To view the console output (cout or printf statements as well as WPILib output) see the next article, [Using Riolog to view console output](#).

## Next Steps

To continue Getting Started with the 2015 Control System, see [Programming your radio for home use](#)

To continue reading about C++, scroll down and click Next to read about [Using Riolog to view console output](#).

## Troubleshooting

This section contains troubleshooting steps for commonly encountered issues.

### Connection Failure

If the deploy process fails to connect to the roboRIO:

- First verify that the roboRIO is powered on, connected to the PC and that the roboRIO is imaged and that your team number is set properly in Eclipse (Window->Preferences->WPILib Preferences).
- The next step is to verify that your computer networking is configured in a way it can communicate with the roboRIO. Visit the roboRIO Network Troubleshooting document in the Troubleshooting section for more details.

# Building and downloading a robot project to the roboRIO

- Check that your firewall is disabled or that an exception has been added for Eclipse. For the Windows Firewall, see Configuring Windows Firewall in the Troubleshooting section for more details.

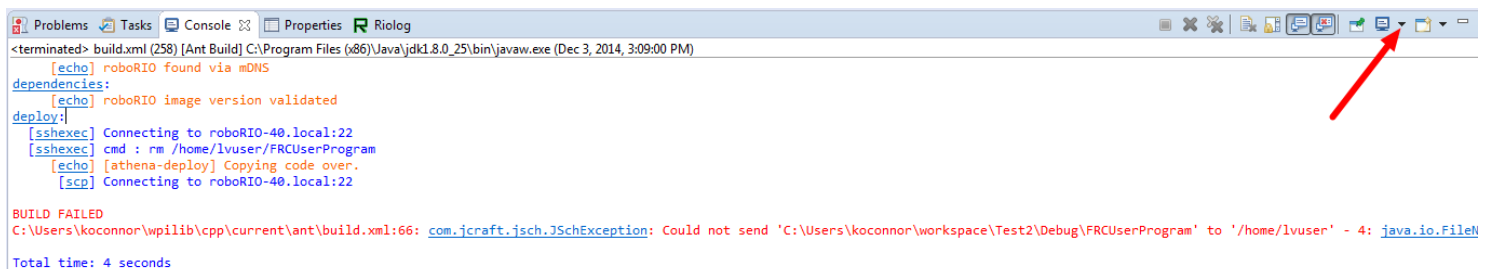
## Image Check Failure

If the Image Check fails that means that the Image version on your roboRIO did not match the image version that the plugins were built against. To check for updates to the Eclipse plugins, make sure you installed the plugins from the web repository and that you have an active internet connection, then click Help->Check For Updates. To check for updates to the NI FRC 2015 Update Suite, which may contain a new roboRIO image, visit the URL from the [Installing the FRC 2015 Update Suite \(All Languages\)](#) document. If an updated version is available, after installing it as described in that document, re-image your roboRIO using the instructions in the [Imaging your roboRIO](#) document.

## sysProps.xml doesn't exist

If you get an error that says something like "wpilib\cpp\current\ant\build.xml:117: Getting Started\sysProps.xml doesn't exist" this indicates that your roboRIO is not imaged or the image is corrupt. Re-image your roboRIO using the instructions in the [Imaging your roboRIO](#) document.

## File Not Found



```
<terminated> build.xml (258) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:09:00 PM)
[echo] roboRIO found via mDNS
dependencies:
[echo] roboRIO image version validated
deploy:
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : rm /home/lvuser/FRCUserProgram
[echo] [athena-deploy] Copying code over.
[scp] Connecting to roboRIO-40.local:22

BUILD FAILED
C:\Users\koconnor\wpilib\cpp\current\ant\build.xml:66: com.jcraft.jsch.JSchException: Could not send 'C:\Users\koconnor\workspace\Test2\Debug\FRCUserProgram' to '/home/lvuser' - 4: java.io.FileN
Total time: 4 seconds
```

If the deploy fails with a "java.io.FileNotFoundException" this means that your program did not build correctly. To see the build failure, either build the program as described in Building a robot program above or click on the dropdown arrow next to the computer monitor icon and select "CDT Global Build Console". The output in the console should provide information about the reason the code failed to build.

## Program "arm-frc-linux-gnueabi-g++" not found in PATH

**Problems** | **Tasks** | **Console** | **Properties** | **Riotlog**

CDT Global Build Console

```
15:19:09 ***** Incremental Build of configuration Debug for project Test2 *****  
Info: Internal Builder is used for build  
arm-frc-linux-gnueabi-g++ -std=c++1y "-IC:\\Users\\koconnor\\wpilib\\cpp\\current\\include" -O0 -g3 -Wall -c -fmessage-length=0 -o "src\\Robot.o" "..\\src\\Robot.cpp"  
Cannot run program "arm-frc-linux-gnueabi-g++": Launching failed  
  
Error: Program "arm-frc-linux-gnueabi-g++" not found in PATH  
PATH=[C:\\Users\\koconnor\\Downloads\\eclipse-cpp-luna-R-win32\\eclipse;C:/Program Files (x86)/Java/jre1.8.0_25/bin/client;C:/Program Files (x86)/Java/jre1.8.0_25/bin;C:/Program Files (x86)/Java/jre1.  
  
15:19:09 Build Finished (took 50ms)
```

If you receive the error "arm-frc-linux-gnueabi-g++" not found in PATH as shown above, it likely means you need to install the C++ toolchains as described in [Installing Eclipse \(C++/Java\)](#). If you have installed the toolchains, look through the PATH entry output by Eclipse and search for an entry that says C:\frc\bin (If it shows up as a drive other than C:\ verify that drive is a permanent drive on your machine, if not you may need to uninstall and re-install the toolchain, making sure to run the installer from a permanent drive). If the entry is not present, try re-installing the toolchains.